In the Claims:

Please amend claims 1, 4, 5, 8, 15, and 18-21 as indicated below.

1. (Currently amended) A system, comprising:

a memory; and

a processor of a server coupled to the memory and configured to execute instructions to:

detect the presence of a mass storage device locally coupled to a stateless client; and

interface said mass storage device to ~~a stateless client that lacks an system, wherein during operation, the stateless client communicates with a server, such that during use, a user interacts with~~ an application executable ~~that executes~~ on the server, ~~by interacting with the stateless client~~ wherein the server is configured to implement a multithreaded execution environment, wherein to interface the mass storage device to the application, the processor is further configured to execute the instructions to implement a device interface through which the mass storage device is visible to the application, and wherein the device interface is implemented by a corresponding device thread that is separately schedulable from other device threads for execution by the server;

~~a mass storage device locally coupled to said stateless client,~~ wherein ~~during operation,~~ said mass storage device is ~~accessed~~ accessible ~~by said user~~ via said server by a user who interacts with said application via said stateless client~~, and wherein the stateless client~~

~~does not locally execute applications that access the mass storage device~~; and

wherein ~~during operation, said server stores~~ the processor is further configured to execute the instructions to store data to said mass storage device via said stateless client in response to said user's interaction with said application.

2. (Previously presented) The system as recited in claim 1, wherein said mass storage device is locally coupled to said stateless client via a Universal Serial Bus (USB) or IEEE 1394 interface.

3. (Previously presented) The system as recited in claim 1, wherein said mass storage device employs magnetic media, optical media, or solid-state storage media.

4. (Currently amended) The system as recited in claim 1, wherein ~~during operation,~~ the server is configured to detect[[s]] a hotplugging event generated in response to disconnection of the mass storage device from the stateless client.

5. (Currently amended) The system as recited in claim 4, wherein in response to detecting the hotplugging event, the server is configured to mark[[s]] resources associated with the mass storage device as deleted or stale, such that disconnection of the mass storage device is visible to the application.

6. (Original) The system as recited in claim 1, wherein said server is further configured to provide a kernel execution mode and a user execution mode, and wherein said server is further configured to execute a storage service daemon, wherein said storage service daemon executes in user execution mode.

7. (Previously presented) The system as recited in claim 1, wherein said mass storage device comprises one or more unit interfaces, wherein each unit interface

comprises one or more logical units (LUNs), and wherein each logical unit comprises one or more partitions.

8. (Currently amended) A method, comprising:

detecting the presence of a mass storage device locally coupled to a stateless client;

interfacing said mass storage device to an application executable on a server, wherein the server is configured to implement a multithreaded execution environment, wherein the server is configured to implement a device interface through which the mass storage device is visible to the application, and wherein the device interface is implemented by a corresponding device thread that is separately schedulable from other device threads for execution by the server;

~~a user interacting with an application that executes on a server, wherein the user interacts with the application via a stateless client that communicates with said server and that lacks an operating system;~~

~~said user accessing a~~ wherein said mass storage device is accessible via said server by a user who interacts with said application via said stateless client, ~~wherein said storage device is locally coupled to said stateless client, and wherein the stateless client does not locally execute user applications that access the mass storage device; and~~

~~said mass storage device~~ storing data to said mass storage device~~, said data being received from said server~~ via said stateless client in response to said user interacting with said application.

9. (Previously presented) The method as recited in claim 8, wherein said mass storage device is locally coupled to said stateless client via a Universal Serial Bus (USB) or IEEE 1394 interface.

10. (Previously presented) The method as recited in claim 8, wherein said mass storage device employs magnetic media, optical media, or solid-state storage media.

11. (Previously presented) The method as recited in claim 8, further comprising disconnecting the mass storage device from the stateless client.

12. (Previously presented) The method as recited in claim 11, further comprising indicating a hotplugging event to the server in response to disconnection of the mass storage device, wherein upon processing of the hotplugging event by the server, disconnection of the mass storage device is visible to the application.

13. (Original) The method as recited in claim 8, wherein said server is further configured to provide a kernel execution mode and a user execution mode, and wherein said server is further configured to execute a storage service daemon, wherein said storage service daemon executes in user execution mode.

14. (Previously presented) The method as recited in claim 8, wherein said mass storage device comprises one or more unit interfaces, wherein each unit interface comprises one or more logical units (LUNs), and wherein each logical unit comprises one or more partitions.

15. (Currently amended) A computer-accessible storage medium that stores program instructions, wherein the program instructions are executable~~, when executed~~ by a server to:

detect the presence of a mass storage device locally coupled to a stateless client; and interface said mass storage device to an application ~~that executes~~ executable on said server, wherein the server is configured to implement[[s]] a multithreaded execution environment, and wherein to interface the mass storage device to the application, the program instructions~~, when executed~~ are executable by the server to [[,]] implement a device interface through which the mass storage device is visible to the application, and wherein the device interface is implemented by a corresponding device thread that is separately schedulable from other device threads for execution by the server;

~~wherein a user interacts with said application via said stateless client,~~ wherein ~~during operation,~~ said mass storage device is ~~accessed~~ accessible ~~by said user~~ via said server by a user who interacts with said application via said stateless client; and

wherein the program instructions[[,]] ~~when executed~~ are executable by the server[[,]] to further store data to said mass storage device via said stateless client in response to said user's interaction with said application.

16. (Previously presented) The computer-accessible storage medium as recited in claim 15, wherein said mass storage device is locally coupled to said stateless client via a Universal Serial Bus (USB) or IEEE 1394 interface.

17. (Previously presented) The computer-accessible storage medium as recited in claim 15, wherein said mass storage device employs magnetic media, optical media, or solid-state storage media.

18. (Currently amended) The computer-accessible storage medium as recited in claim 15, wherein to detect the presence of the mass storage device, the program

instructions~~, when executed~~ are executable by the server[[,]] to detect a hotplugging event generated in response to coupling of the mass storage device to the stateless client.

19. (Currently amended) The computer-accessible storage medium as recited in claim 18, wherein in response to detecting the hotplugging event, the program instructions~~, when executed~~ are executable by the server[[,]] to create the device interface through which the mass storage device is visible to the application.

20. (Currently amended) The computer-accessible storage medium as recited in claim 15, wherein said server provides a kernel execution mode and a user execution mode, and wherein said program instructions the program instructions~~, when executed~~ are executable by the server[[,]] to implement:

a storage service daemon, wherein said storage service daemon executes in user execution mode;

a kernel interface that executes in kernel execution mode and ~~provides~~ is configured to provide to applications that execute on the server a public interface to the mass storage device coupled to the stateless client; and

a daemon interface that executes in kernel execution mode and ~~conveys~~ is configured to convey device access requests from the kernel interface to the storage service daemon, wherein the interface between daemon interface and the storage service daemon is private with respect to applications that execute on the server.

21. (Currently amended) The computer-accessible storage medium as recited in claim 20, wherein the storage service daemon comprises:

a plurality of device threads, each corresponding to a respective device interface, and each device thread being separately schedulable from other device threads for execution by the server;

a remote disk driver that, upon receiving an access request from a particular device thread, is configured to assemble[[s]] a command for the mass storage device according to a command set implemented by the mass storage device;

a protocol driver that is configured to assemble[[s]] one or more commands received from the remote disk driver into a transaction for the mass storage device according to a peripheral interface with which the mass storage device is coupled to the stateless client.